



# Open Security Controls Assessment Language (OSCAL) Leveraged Authorizations and Customer Responsibilities

Brian J. Ruf, CISSP, CCSP, PMP

National Institute of Standards and Technology

Version 4

September 4, 2020

# Three Scenarios

## ► Scenario 1: OSCAL SSP / With Access

- The leveraged system is using an OSCAL SSP; and the leveraging system is permitted to access it.
- No CRM is needed.
- **Preferred approach!**

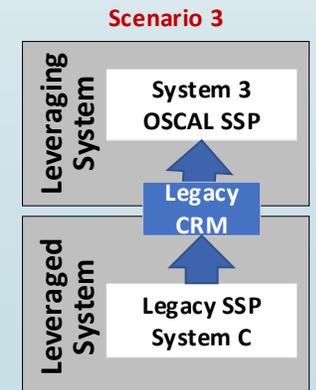
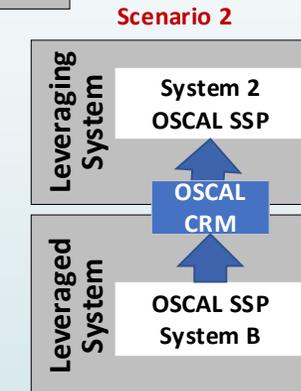
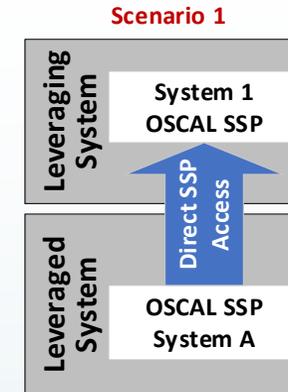
Today's Focus

## ► Scenario 2: OSCAL SSP / No Access

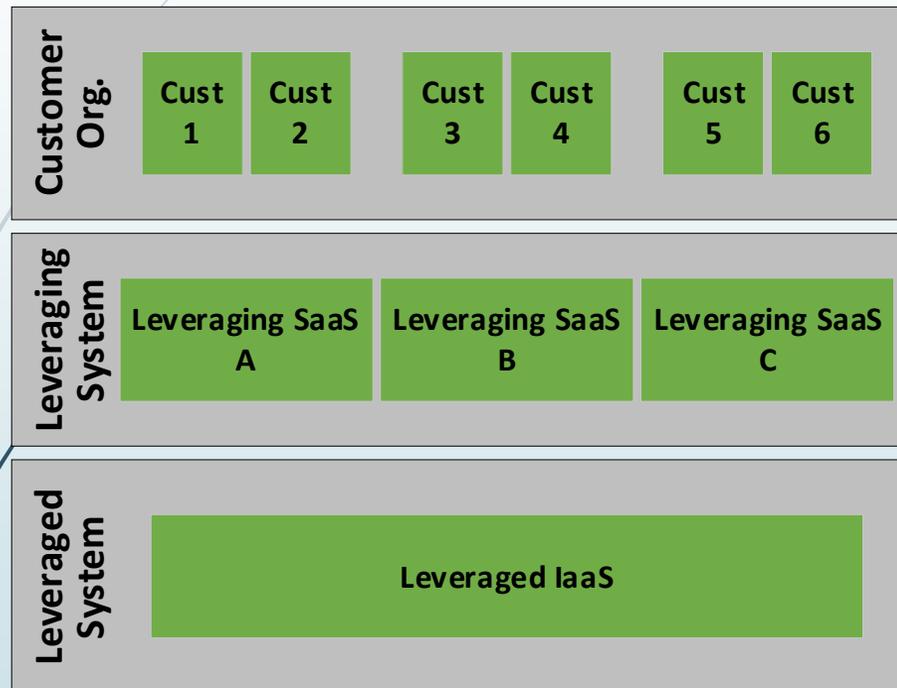
- The leveraged system is using an OSCAL SSP; however, the leveraging system is not permitted access it.
- An OSCAL CRM is used.

## ► Scenario 3: Legacy SSP

- A leveraged system is still using a legacy SSP.
- A legacy Customer Responsibility Matrix (CRM) is used.

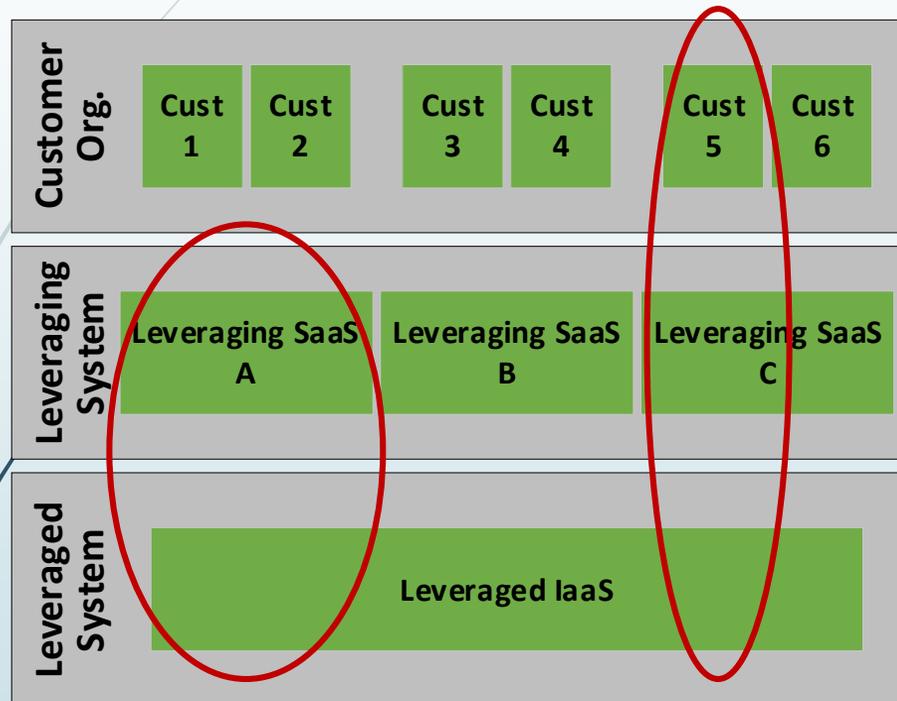


## What is a Leveraged Authorization (LA)?



- **A leveraged authorization exists where:**
  - one or more leveraging systems relies on a leveraged for operation; and
  - any leveraging system is authorized separately from the leveraged system.
- Common examples:
  - **Cloud:** Several SaaS systems running on a separately authorized IaaS.
  - **Legacy:** Several systems relying on a separately authorized storage array or other general support system (GSS)
- Leveraged Authorization vs Interconnection
  - A leveraged authorization is more of a hierarchical relationship
  - An interconnection is more of a peer relationship

# Responsibility and Adjudication

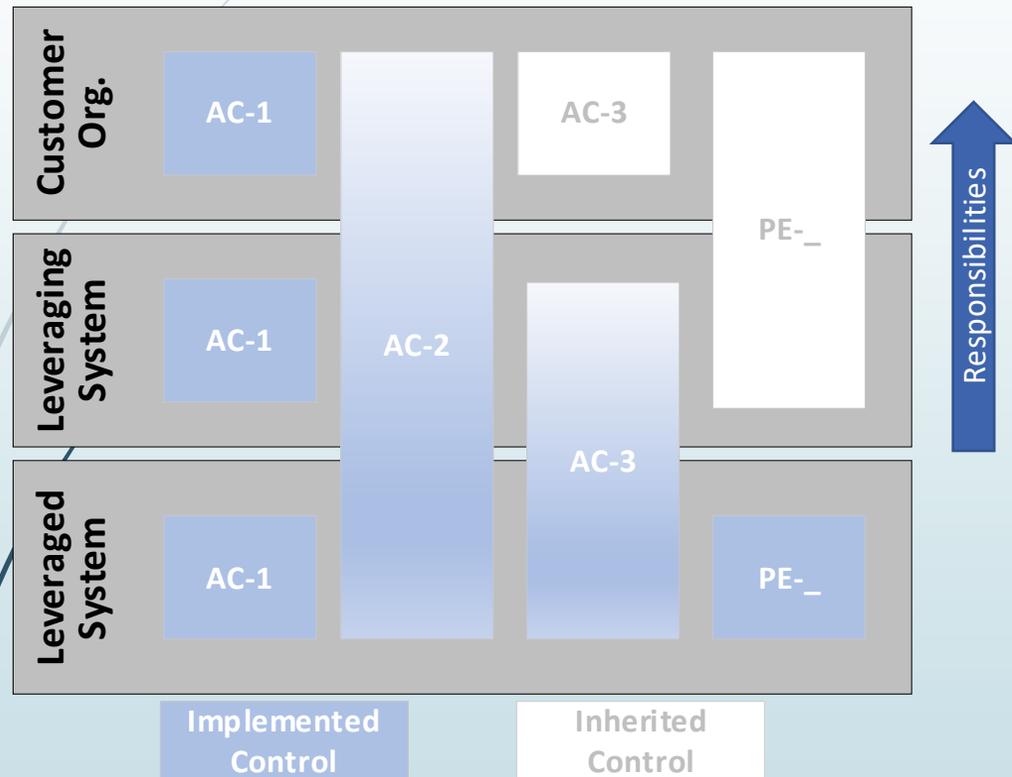


- An authorizing official (AO) must adjudicate the entire stack - relative to the authorization they are issuing. (Holistic View)
- Each system owner is responsible for their system in the stack. (System View)

## Examples:

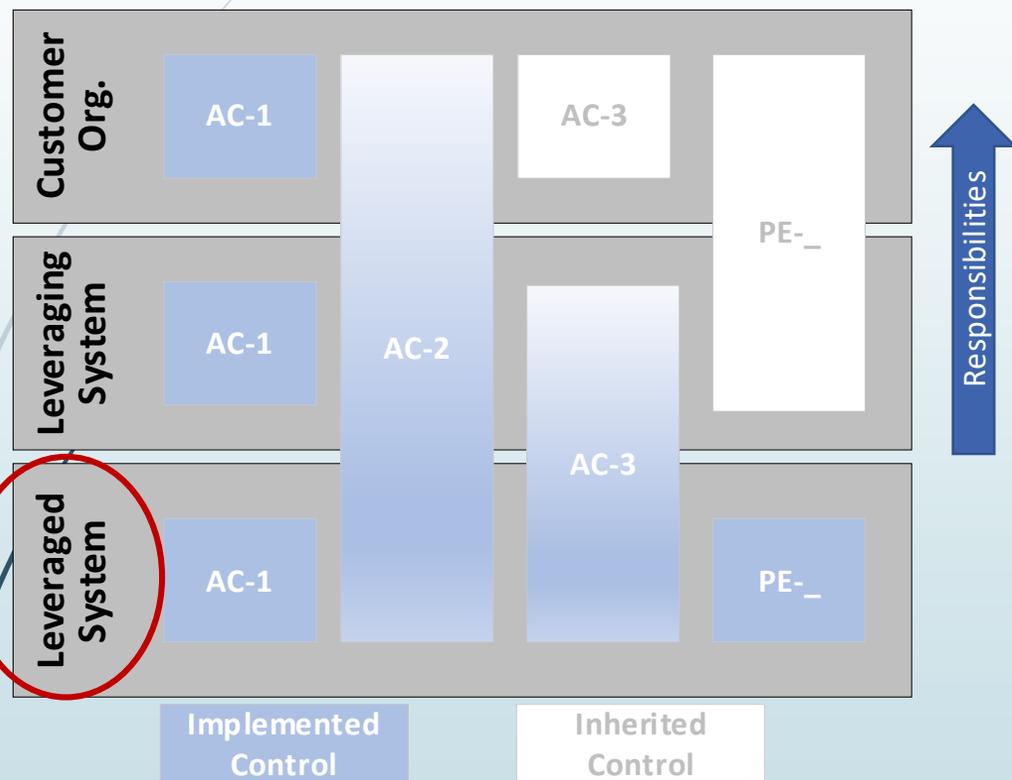
- The AO for *Leveraging SaaS A* must adjudicate its authorization in consideration of the controls within both:
  - *Leveraging SaaS A*; and
  - the *Leveraged IaaS*.
- The AO for *Cust 5* must adjudicate its authorization in consideration of the controls implemented:
  - by the *Cust 5*;
  - within the *Leveraging SaaS C*; and
  - within the *Leveraged IaaS*.

## Full Control Satisfaction (Holistic View)



- ▶ **Some controls must be satisfied independently by each system**
  - ▶ Example: FedRAMP does not allow inheritance for XX-1 controls.
- ▶ **Some controls are only fully satisfied if each system does their part.**
  - ▶ Example: Logical access control must be implemented on all components in “the stack”.
- ▶ **Some controls are fully satisfied at a lower level, thus fully inherited higher in the stack.**
  - ▶ Example: Usually an IaaS takes care of all physical controls. Each SaaS has no ability to implement physical controls and fully inherits those controls from the IaaS.

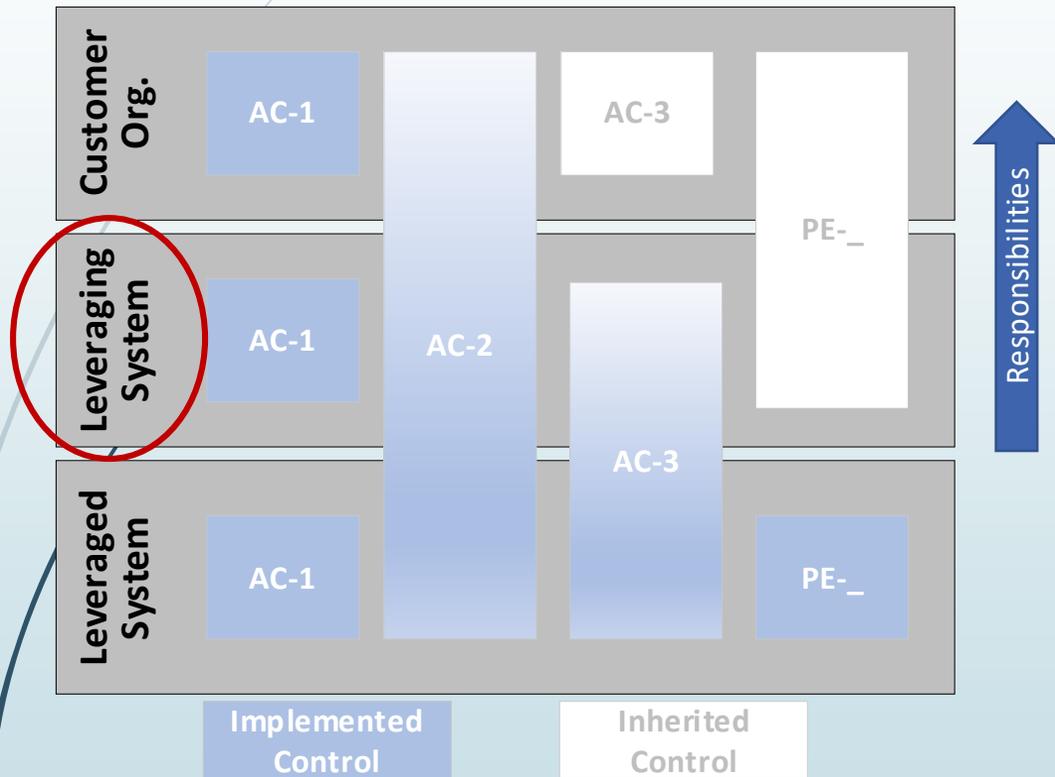
## Full Control Satisfaction (System View)



### ► Leveraged System:

- The leveraged system may address each control, through some combination of:
  - implementing the control; or
  - defining a customer responsibility for the control.

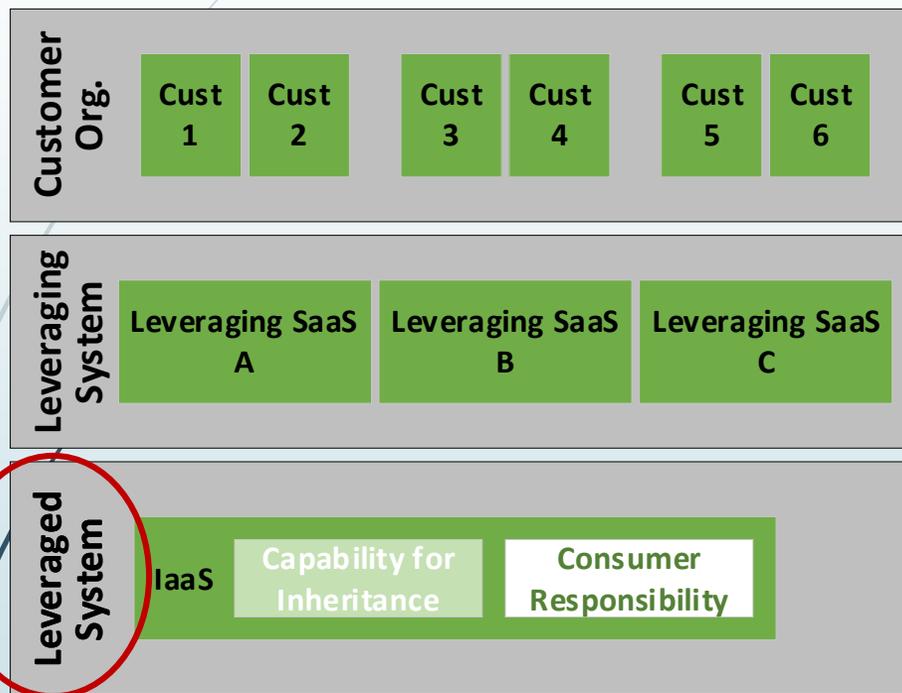
## Full Control Satisfaction (System View)



### ► Leveraging System:

- The leveraging system may address each control, through some combination of:
  - **identifying an inherited control from a leveraged system;**
  - implementing the control; or
  - defining a customer responsibility for the control

## Control Documentation (System View)



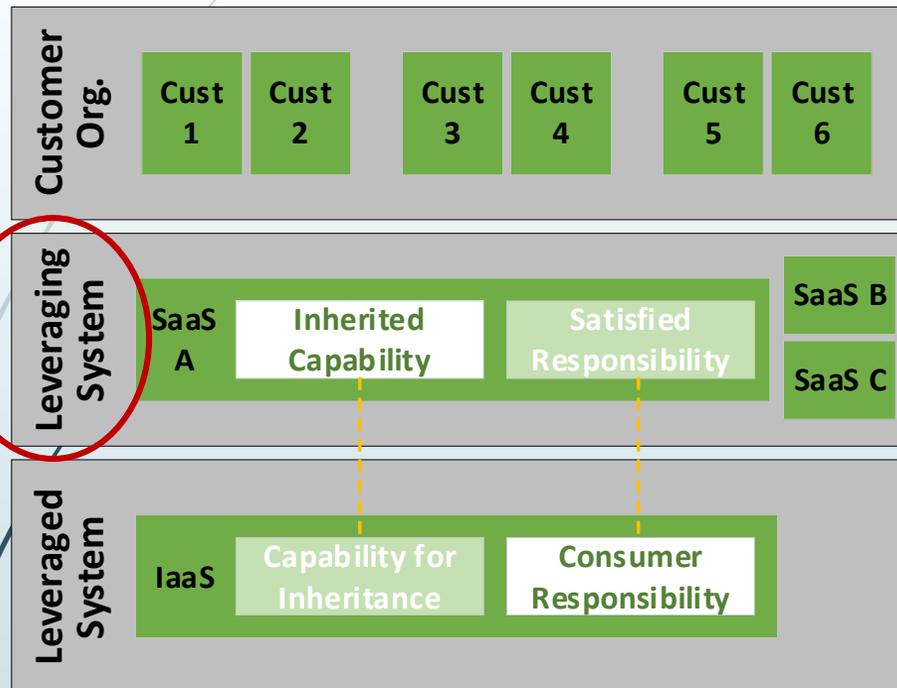
### ► Leveraged System:

- The leveraged system's SSP should:
  - identify what **may be** inherited by leveraging systems
    - including a consumer-appropriate description of the control inheritance
  - any responsibilities the leveraging system must address to fully satisfy a control ...

... including where:

- the leveraging system must configure an inherited capability; or
- there is a gap in control satisfaction which must be addressed by the leveraging system

## Control Documentation (System View)



### ► Leveraging System:

- The leveraging system's SSP should:
  - identify what **is** inherited from a leveraged system
  - identify any addressed responsibilities (as identified by the leveraged system)

# Looking at the OSCAL

10

## Leveraged System (Inheritance - associate with each component providing an inheritable capability)

```
<control-implementation>
  <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accd-d3e534932efb" />
  <implemented-requirement control-id="ac-2" uuid="uuid-value">
    <annotation name="implementation-status" ns="https://fedramp.gov/ns/oscal" value="implemented" />
    <responsible-role role-id="admin-unix"/>
    <responsible-role role-id="program-director"/>
    <set-parameter param-id="ac-2_prm_1"><value>[SAMPLE]privileged, non-privileged</value></set-parameter>

    <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

      <by-component component-uuid="uuid-of-component-this-system" uuid="uuid-value">
        </by-component>

      <by-component component-uuid="uuid-of-component-application" uuid="uuid-value">
        <description>
          <p>Description of how the application component satisfies AC-2, part a.</p>
        </description>

        <annotation name="inheritance" value="consumer" uuid="uuid-value">
          <remarks>
            <p>Consumer-appropriate description of what may be inherited.</p>
            <p>In the context of the application component in satisfaction of AC-2, part a.</p>
          </remarks>
        </annotation>

        <annotation name="responsibility" value="consumer" uuid="uuid-value">
          <remarks>
            <p>Leveraging system's responsibilities with respect to inheriting this capability.</p>
            <p>In the context of the application component in satisfaction of AC-2, part a.</p>
          </remarks>
        </annotation>

      </by-component>
    </statement>
  </implemented-requirement>
```

# Looking at the OSCAL

11

## Leveraged System (Customer responsibility to address gap - associate with the "This System" component)

```
<control-implementation>
  <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-acc-d3e534932efb" />
  <implemented-requirement control-id="ac-2" uuid="uuid-value">
    <annotation name="implementation-status" ns="https://fedramp.gov/ns/oscal" value="implemented" />
    <responsible-role role-id="admin-unix"/>
    <responsible-role role-id="program-director"/>
    <set-parameter param-id="ac-2_prm_1"><value>[SAMPLE]privileged, non-privileged</value></set-parameter>

    <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

      <by-component component-uuid="uuid-of-component-this-system" uuid="uuid-value">
        <description>
          <p>Description of how AC-2, part a is satisfied within this system.</p>
        </description>

        <annotation name="responsibility" value="consumer" uuid="uuid-value">
          <remarks>
            <p>Leveraging system's responsibilities with respect to fully satisfying AC-2, part a.</p>
          </remarks>
        </annotation>
      </by-component>

    </statement>
  </implemented-requirement>
```

# Looking at the OSCAL

12

## Leveraging System (inheriting the capability of a component)

```
<control-implementation>
  <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
  <implemented-requirement control-id="ac-2" uuid="uuid-value">
    <!-- details cut -->
    <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

      <by-component component-uuid="uuid-of-component-this-system" uuid="uuid-value">
        </by-component>

        <by-component component-uuid="uuid-of-component-application-from-leveraged-ssp" uuid="uuid-value">
          <description>
            <p>Leveraging system's custom description of the implemented capability.</p>
            <p>In the context of the leveraged system's application, as it satisfies AC-2, part a.</p>
          </description>

          <annotation name="inherited" value="uuid-of-leveraged-inheritance">
            <remarks>
              <p>Original inheritance description from leveraged system.</p>
            </remarks>
          </annotation>

          <annotation name="satisfied-responsibility" value="uuid-of-leveraged-consumer-responsibility">
            <remarks>
              <p>Describe how this responsibility (configuration) was satisfied for AC-2, part a.</p>
            </remarks>
          </annotation>

        </by-component>

      </statement>
    </implemented-requirement>
```

# Looking at the OSCAL

13

## Leveraging System (consumer responsibility - fully addressing a gap)

```
<control-implementation>
  <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-acco-d3e534932efb" />
  <implemented-requirement control-id="ac-2" uuid="uuid-value">
    <!-- details cut -->
    <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

      <by-component component-uuid="uuid-of-component-this-system" uuid="uuid-value">
        </by-component>

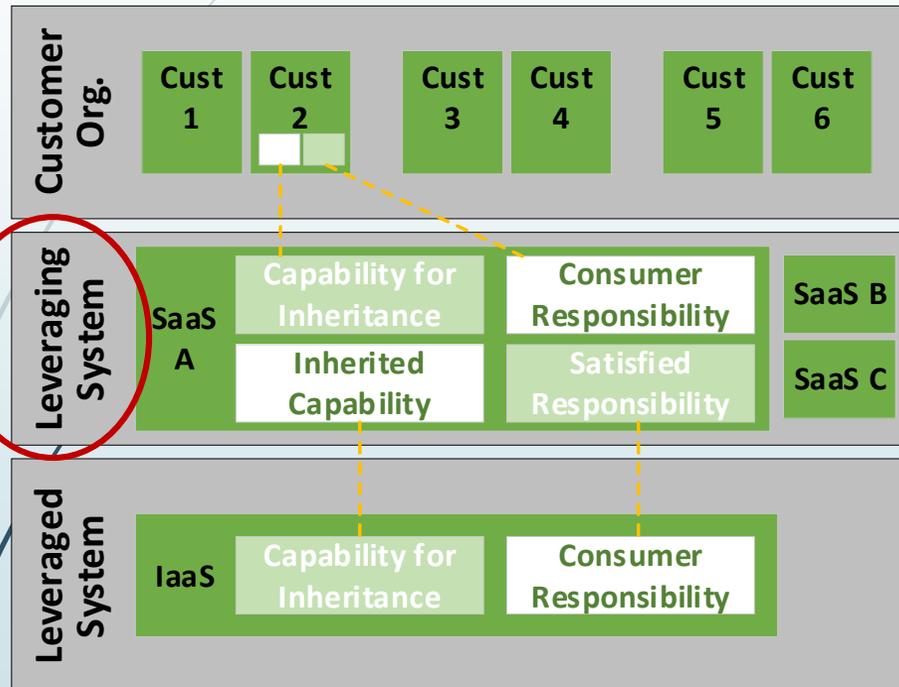
      <by-component component-uuid="uuid-of-component-this-system-from-leveraged-ssp" uuid="uuid-value">
        <description>
          <p>Leveraging system's custom description of the implemented capability.</p>
        </description>

        <annotation name="satisfied-responsibility" value="uuid-of-leveraged-consumer-responsibility">
          <remarks>
            <p>Describe how this responsibility (gap) was satisfied for AC-2, part a.</p>
          </remarks>
        </annotation>

      </by-component>

    </statement>
  </implemented-requirement>
```

## Control Documentation (System View)



### ► Leveraging System:

- The leveraging system's SSP should:
  - identify what **is** inherited from a leveraged system
  - identify any addressed responsibilities (as identified by the leveraged system)

In addition to:

- identifying what **may be** inherited by the leveraging system's customers
- any responsibilities the leveraging system's customers must address to fully satisfy a control

# Looking at the OSCAL

15

## Leveraging System (consumer responsibility - partially addressing a gap)

```
<control-implementation>
  <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-acc-d3e534932efb" />
  <implemented-requirement control-id="ac-2" uuid="uuid-value">
    <!-- details cut -->
    <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

      <by-component component-uuid="uuid-of-component-this-system" uuid="uuid-value">
        </by-component>

      <by-component component-uuid="uuid-of-component-this-system-from-leveraged-ssp" uuid="uuid-value">
        <description>
          <p>Leveraging system's custom description of the implemented capability.</p>
        </description>

        <annotation name="satisfied-responsibility" value="uuid-of-leveraged-consumer-responsibility">
          <remarks>
            <p>Describe how this responsibility (gap) was satisfied for AC-2, part a.</p>
          </remarks>
        </annotation>

        <annotation name="responsibility" value="consumer" uuid="new-uuid-value">
          <remarks>
            <p>Revised remaining consumer responsibility.</p>
          </remarks>
        </annotation>

      </by-component>

    </statement>
  </implemented-requirement>
```

# Looking at the OSCAL

16

## Leveraging System (consumer responsibility - passing downstream as-is)

```
<control-implementation>
  <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-acc-d3e534932efb" />
  <implemented-requirement control-id="ac-2" uuid="uuid-value">
    <!-- details cut -->
    <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

      <by-component component-uuid="uuid-of-component-this-system" uuid="uuid-value">
        </by-component>

      <by-component component-uuid="uuid-of-component-this-system-from-leveraged-ssp" uuid="uuid-value">
        <description>
          <p>Leveraging system's decision/rationale for fully passing responsibility downstream.</p>
        </description>

        <annotation name="responsibility" value="consumer" uuid="uuid-from-leveraged-ssp">
          <remarks>
            <p>Leveraging system's responsibilities with respect to fully satisfying AC-2, part a.</p>
          </remarks>
        </annotation>

      </by-component>

    </statement>
  </implemented-requirement>
```

# Questions? Thank you!

**We want your feedback!**

**OSCAL Repository:**

<https://github.com/usnistgov/OSCAL>

**Project Website:**

<https://www.nist.gov/oscal>

**How to Contribute:**

<https://pages.nist.gov/OSCAL/contribute/>

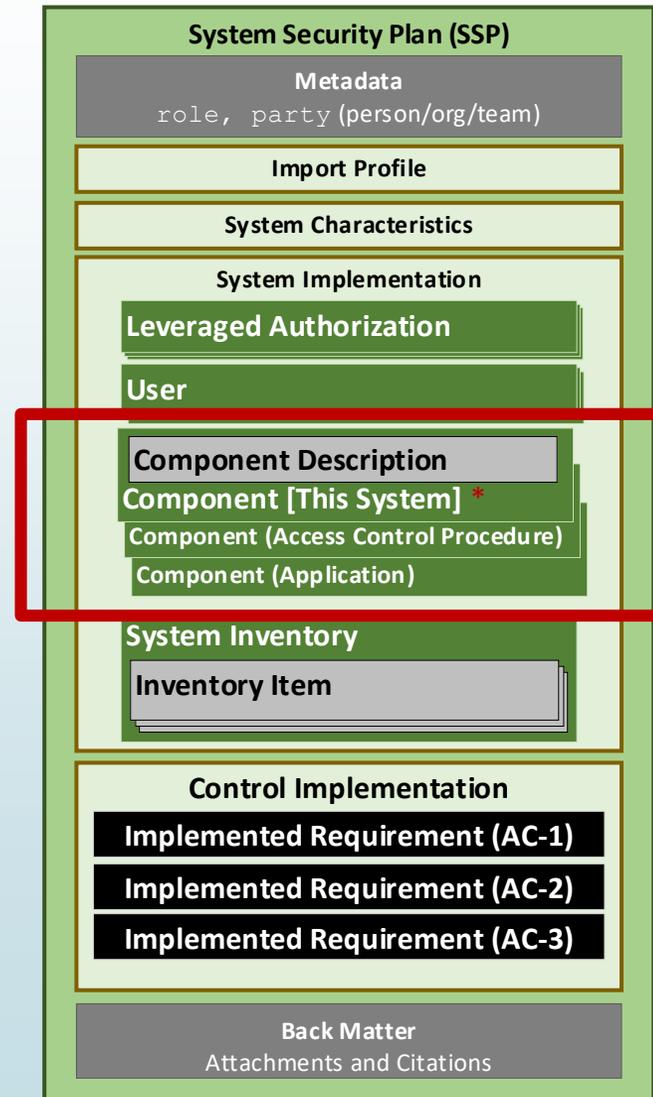
**FedRAMP Implementation Guides**

<https://github.com/gsa/fedramp-automation> (Available in July)

BACKUP SLIDE(S)

# Responding to Controls in the SSP: Define Components

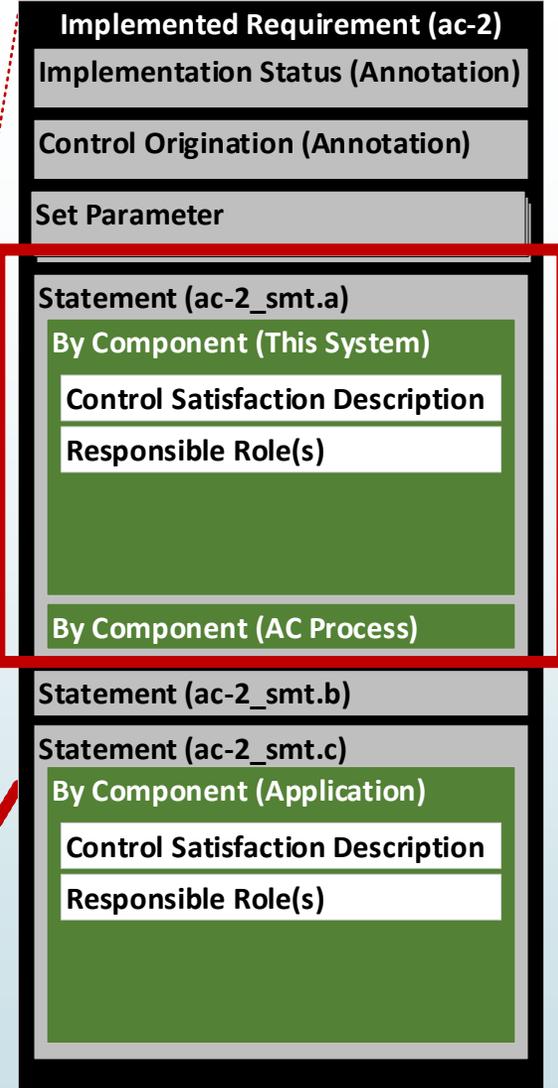
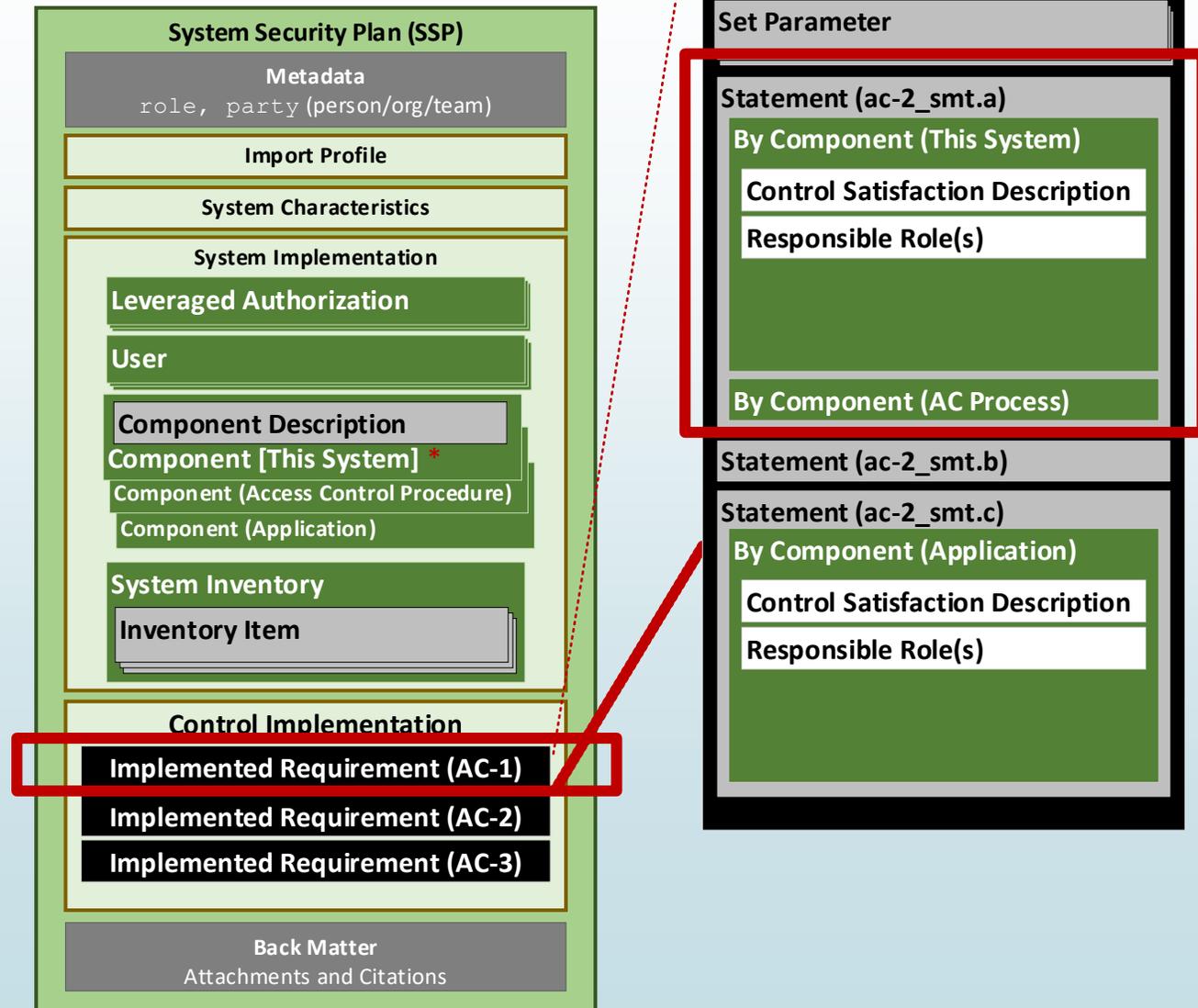
- Each control response is broken down to the individual components involved.
- Enables a more robust response to controls
- Example: The access control implementation that satisfies AC-2, part a is described separately for:
  - This System
  - The Access Control Procedure
  - A shared Application



- Components are defined in the `system-implementation` assembly.
- One `component` assembly for each component.
- There must always be a "This System" component defined.
- Other components are defined as appropriate.
- SSP authors have flexibility in how granular they define components.

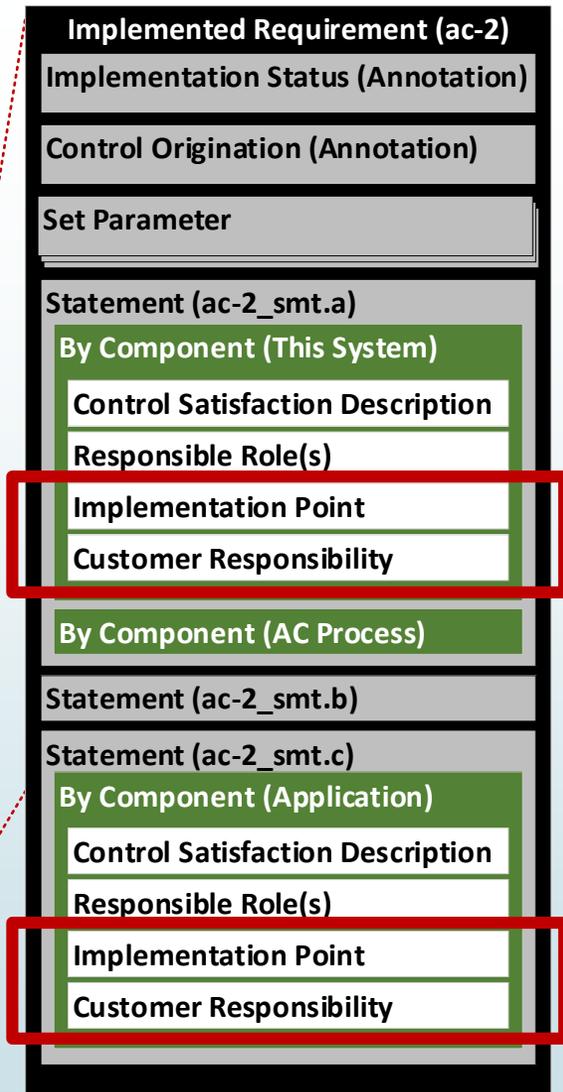
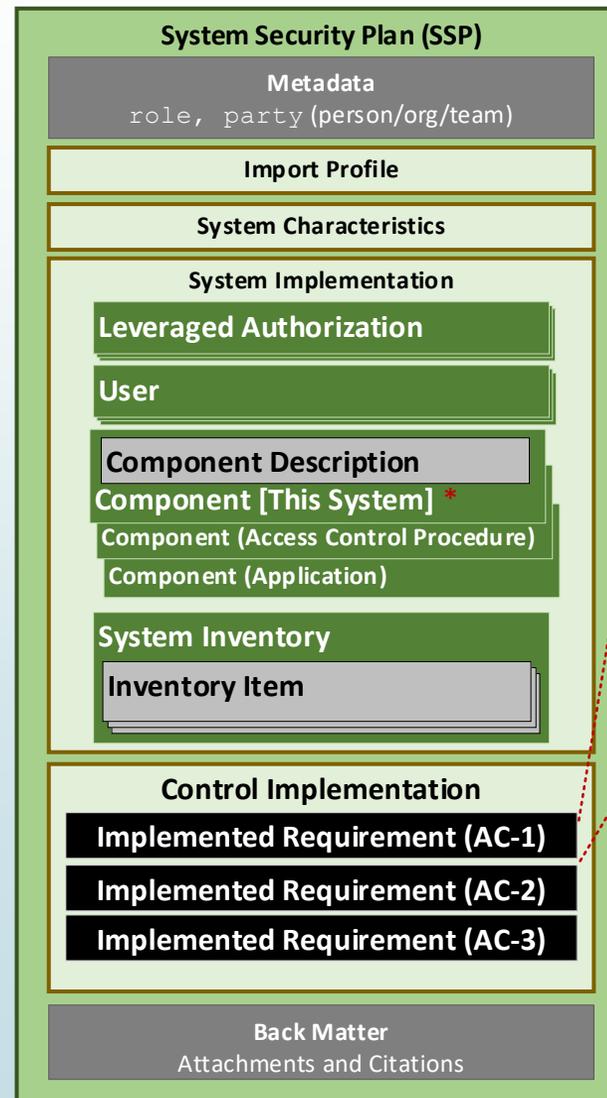
# Responding to Controls in the SSP: Respond By Component

- For each control there is an implemented-requirement assembly.
- Within each implemented-requirement assembly, there are one or more statement assemblies.
- Each statement assembly has one or more by-component assemblies. Each references a component involved with control satisfaction.
- Control satisfaction responses are provided in the description field within each by-component assembly.
- NOTE: Use the "This System" component for any control satisfaction explanation that does not fit cleanly with a more specific component, or to describe how the components work together.



# Correct Placement of Customer Responsibility Statements

- Customer responsibility statements are placed within applicable `by-component` assembly using an `annotation`.
- If the customer has a responsibility within the application, there should be a `by-component` assembly in the statement assembly, which identifies the application and includes the customer responsibility `annotation`.
- If a customer responsibility statement does not fit any specific component, place it in the "This System" component.



# Looking at the OSCAL (Components)

22

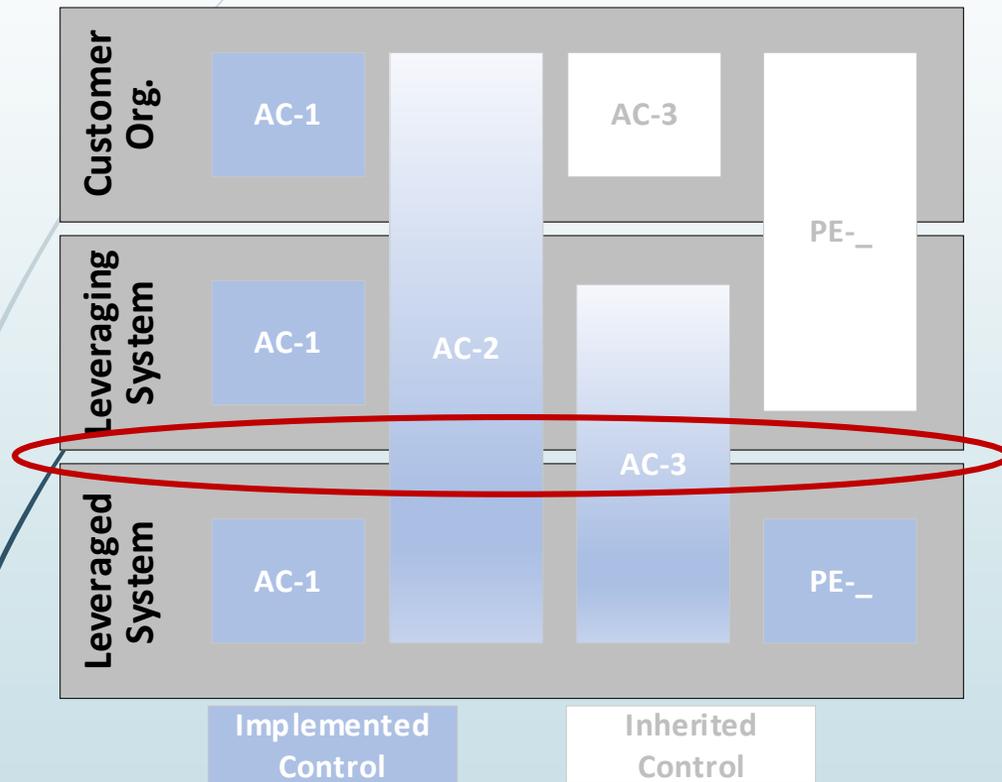
## Leveraged System

```
<system-implementation>
  <user />
  <component uuid="11111111-0000-4000-9001-000000000001" component-type="system">
    <title>This System</title>
    <description>
      <p>This Leveraged IaaS.</p>
      <p>The entire system as depicted in the system authorization boundary</p>
    </description>
    <status state="operational"/>
  </component>

  <component uuid="11111111-0000-4000-9001-000000000002" component-type="procedure">
    <title>Access Control Procedure</title>
    <description>
      <p>This is the procedure that governs access to the application.</p>
    </description>
    <link href="#8b9d82a9-dd49-4309-a466-685b0081f28c"/>
    <status state="operational"/>
  </component>

  <component uuid="11111111-0000-4000-9001-000000000003" component-type="software">
    <title>Application</title>
    <description>
      <p>An application within the IaaS, exposed to SaaS customers and their downstream customers.</p>
      <p>This Leveraged IaaS maintains aspects of the application.</p>
      <p>The Leveraging SaaS maintains aspects of their assigned portion of the application.</p>
      <p>The customers of the Leveraging SaaS maintain aspects of their sub-assigned portions of the application.</p>
    </description>
    <status state="operational"/>
    <responsible-role role-id="admin">
      <party-uuid>11111111-0000-4000-9000-100000000001</party-uuid>
    </responsible-role>
  </component>
</system-implementation>
```

## Relationship View: SSP Documentation



### ► The Leveraged System's SSP:

- may provide information about controls that may be inherited by a leveraging system
- must explicitly identify all customer responsibilities required to fully satisfy a control
  - The number of levels beyond the leveraging system is irrelevant

### ► The Leveraging System's SSP:

- must identify what is inherited from the leveraged system
- must address control requirements not explicitly satisfied through inheritance
- should link customer responsibilities identified by its leveraged system to:
  - control implementation statements
  - customer responsibilities the leveraging system defined for its downstream customers

# Leveraged System -> Leveraging System

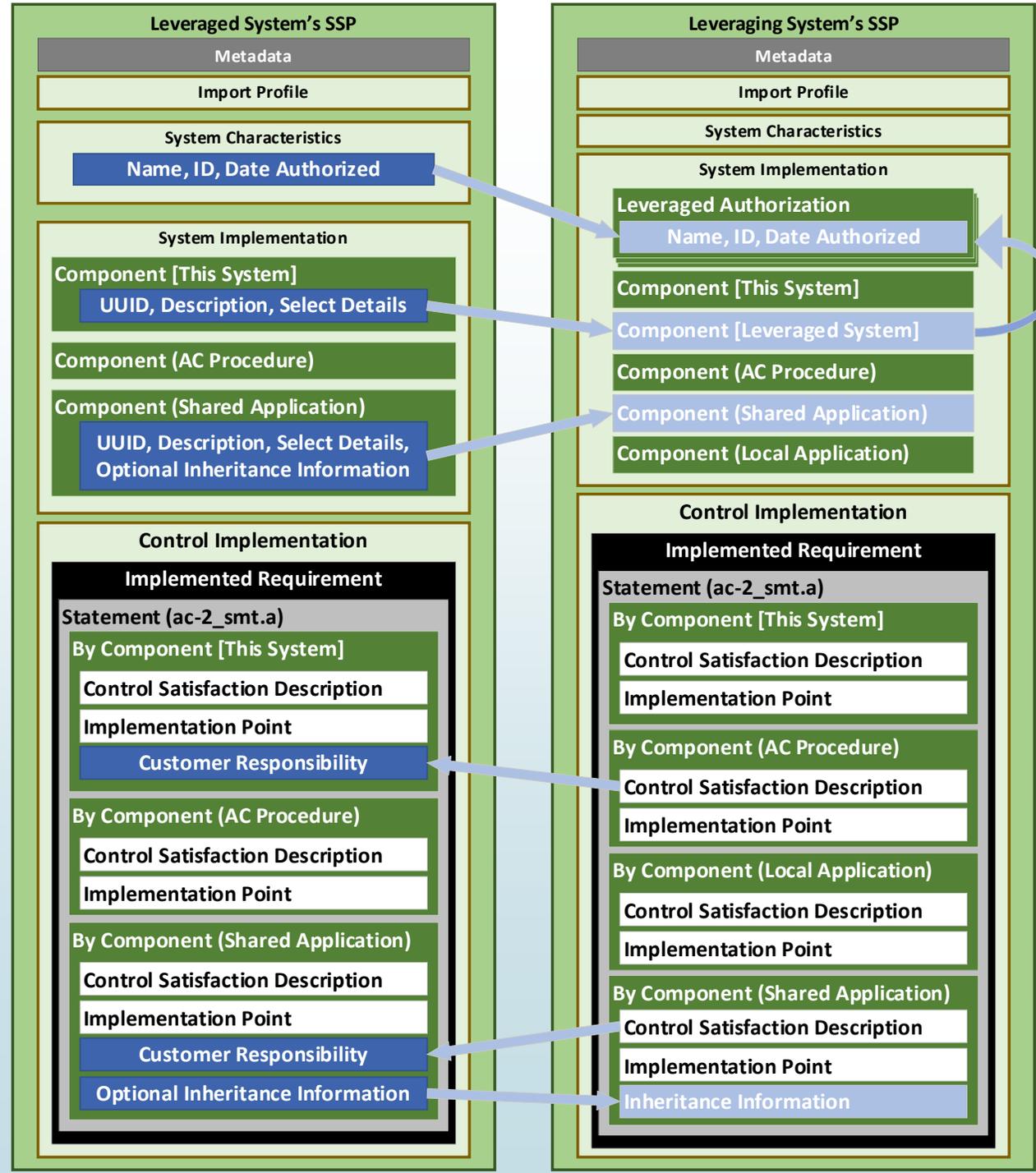
## Use Cases

- The Leveraged System has an application exposed to the Leveraging System
  - The customer configuration responsibilities are defined within *AC-2, part a*; within a by-component assembly associated with the application
  - An optional inheritance statement is defined within *AC-2, part a*; within a by-component assembly associated with the application. It describes additional aspects of *AC-2, part a* addressed by the application with no customer requirement.
  - The component definition for the application is communicated to the leveraging system
- The Leveraged System has an access control procedure
  - The procedure is only for the leveraged system. The leveraging system requires its own procedure to satisfy *AC-2, part a*.
  - A customer responsibility statement is made with within *AC-2, part a*; within a by-component assembly associated with "This System" describing the need for the customer to create their own access control procedure.
  - In this instance it does not make sense to include the component representing the leveraged system's access control procedure.

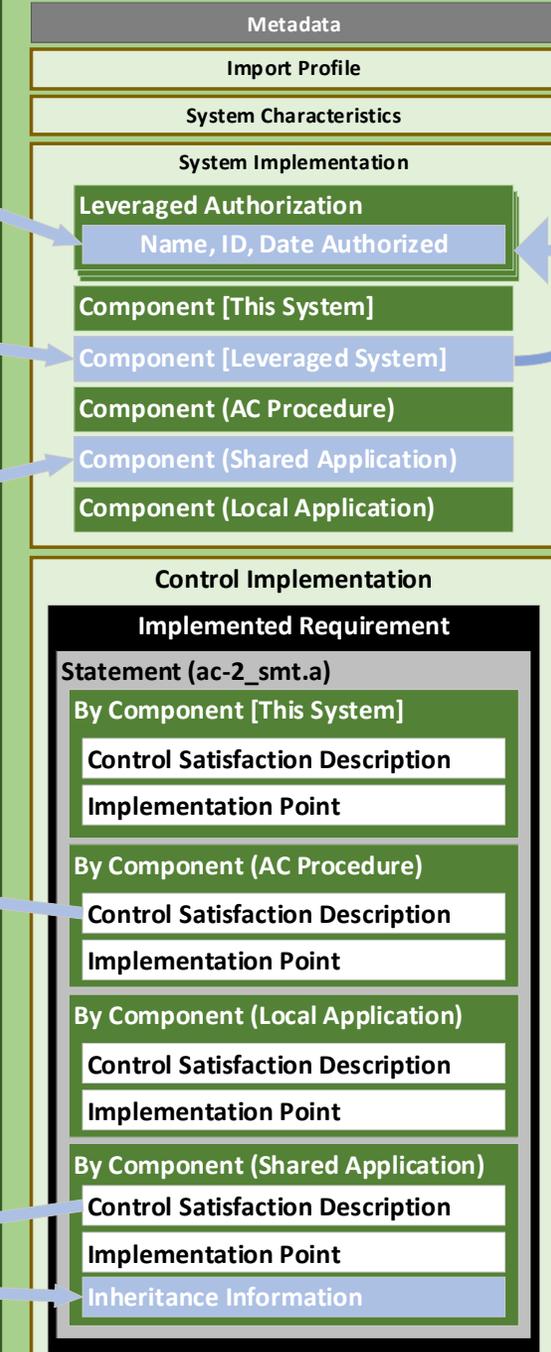
# Leveraging System

**A leveraging system must communicate the following to customers and AOs:**

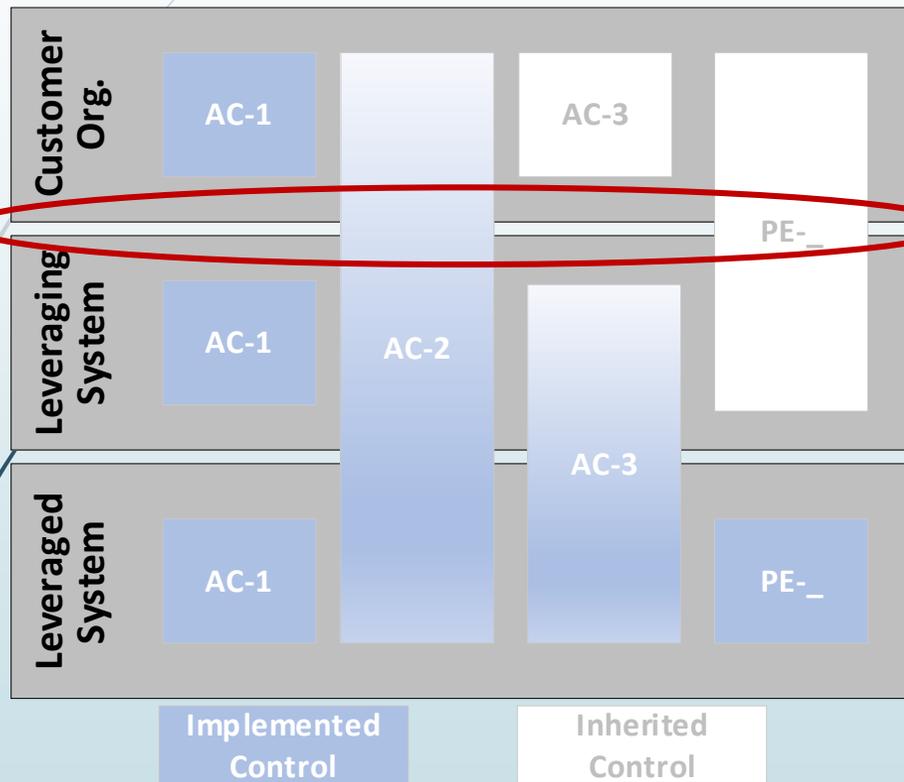
- Information about the authorizations for both the Leveraging and Leveraged Systems (dates, system IDs, etc.)
- Control Satisfaction Descriptions that satisfy a customer responsibility statement
- Statements about what the leveraging system has inherited from the leveraged system
  - In the component definition; and/or
  - In the by-component response to a specific control/part
- Component information from the leveraged system must be referenced in the leveraging system
- End Consumer (Customer) responsibility statements may also be defined the same way the leveraged system defines them



## Leveraging System's SSP



## Relationship Views: Simplify and Modularize



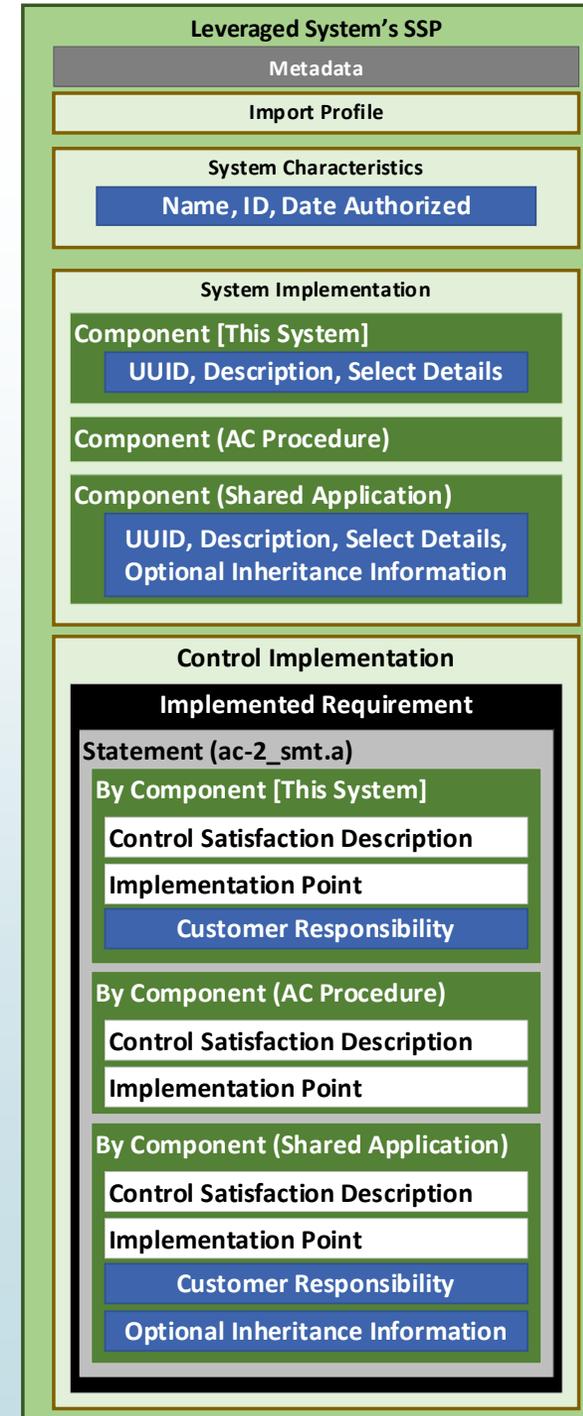
### ► For additional layers:

- The leveraging system becomes the leveraged system relative to the customer layer
- In addition to
  - information controls that may be inherited by a leveraging system
  - explicit customer responsibilities required to fully satisfy a control
- The number of levels beyond the leveraging system is irrelevant.

# Leveraged System

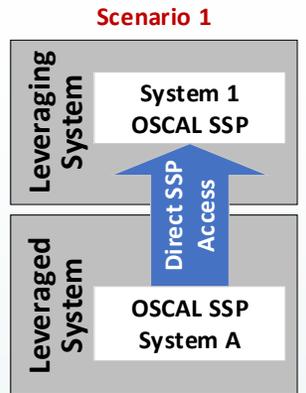
**A leveraged system must communicate the following to a leveraging system:**

- Information about the Leveraged System's authorization (date, system ID, etc.)
- Consumer (Customer) responsibility statements
  - In the by-component response to a specific control/part
    - System-wide statements - associated with the by-component statement for "This System"
    - Component-specific statements
- Statements about what the leveraging system could inherited
  - In the component definition; and/or
  - In the by-component response to a specific control/part
- Certain information about any component associated with consumer responsibility or inheritance statements



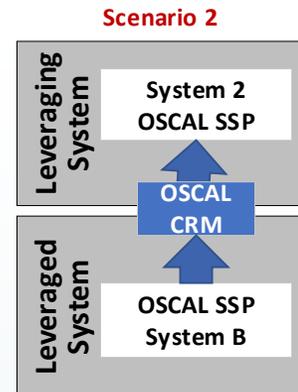
# Scenario 1: OSCAL SSP With Access

- Preferred scenario
- The SSP of the **leveraging system** can "see" the **leveraged system's** SSP
- Tools can identify which statements in the **leveraged system's** SSP have a customer responsibility
- Tools can further identify the **leveraged system's** components associated with these customer responsibility statements.
- The **leveraging system's** ISSO must determine if fulfillment of their customer responsibility involves the component from the **leveraged system**, or a new component that must be supplied by the **leveraging system's** organization.

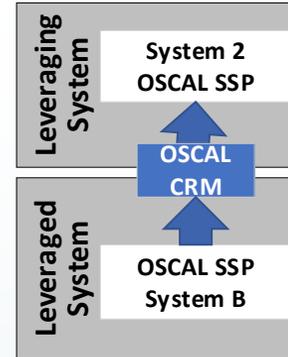
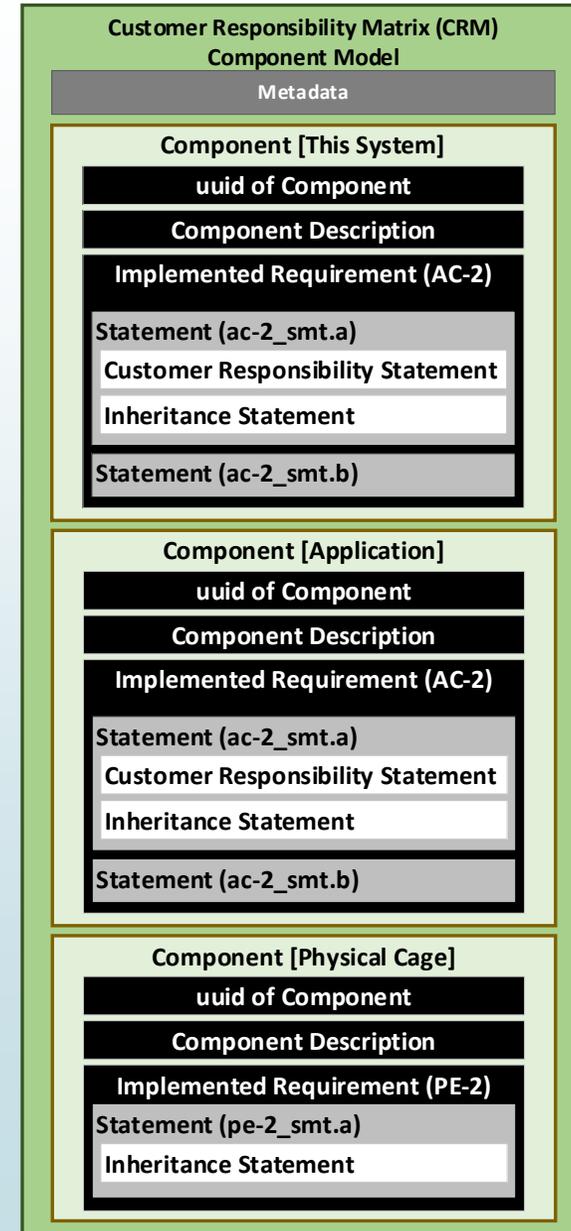
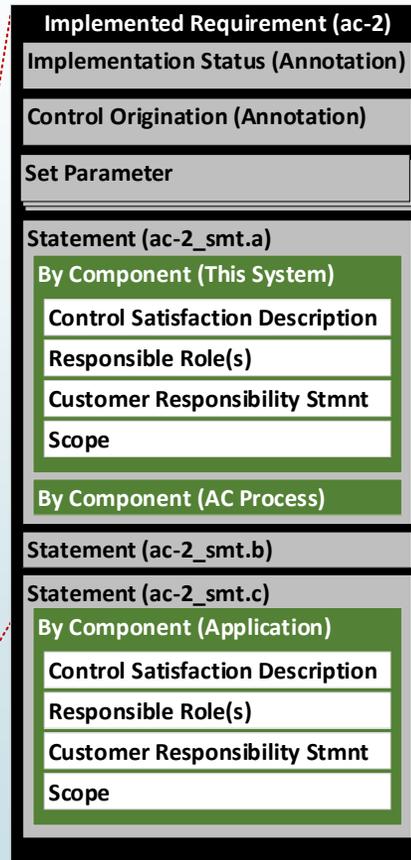
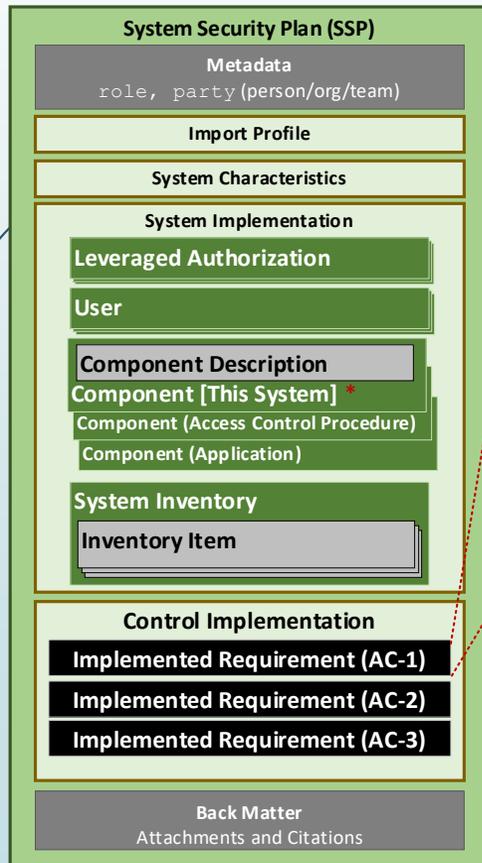


## Scenario 2: OSCAL SSP - No Access

- ▶ The SSP of the **leveraging system** is not permitted to "see" the full **leveraged system's** SSP.
- ▶ The **leveraged system's** owner, creates an OSCAL customer responsibility matrix (CRM), using the OSCAL Component model.
- ▶ Every component in the **leveraged system's** SSP, with a customer responsibility annotation is created in the OSCAL CRM with only basic information, such as the component title and general description.
  - ▶ The exact level of detail is a situation-specific decision.
  - ▶ The original Component UUID value from the **leveraged system's** SSP must be duplicated.
  - ▶ Every control, which cites that component AND associates it with a customer responsibility statement is cited in the control-implementation assembly within the component.
  - ▶ The entire "responsibility" annotation is duplicated from the SSP model by-component entry to the Component model statement-id assembly.
- ▶ The **leveraging system's** ISSO must determine if fulfillment of their customer responsibility involves the component from the **leveraged system**, or a new component that must be supplied by the **leveraging system's** organization.
  - ▶ If the **leveraged system's** component is used, the **leveraging system's** SSP must import the component detail from the CRM into the leveraging system's SSP.
  - ▶ The original UUID must be maintained.
  - ▶ The **leveraging system's** SSP must ensure they fully satisfy every customer responsibility statement in the CRM, which requires at least one entry within the cited statement.

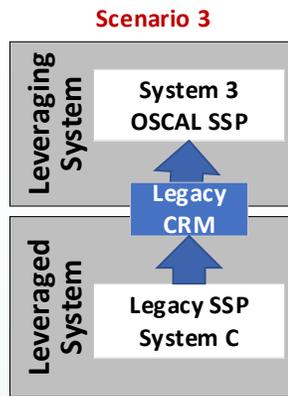


# Scenario 2: OSCAL SSP: No Access



## Scenario 3: Legacy SSP or CRM

- ▶ The **leveraged system's** SSP is not expressed in OSCAL, or its CRM is not.
- ▶ The **leveraging system** SSP must define an additional component representing the **leveraged system** itself.
- ▶ Every responsibility statement in the **leveraged system's** legacy SSP/CRM must be addressed by the **leveraging system's** SSP within the cited control statement.
- ▶ If the responsibility is addressed by customer action in the **leveraged system**, the **leveraging system's** statement should cite that component. Otherwise, it should cite the appropriate component.



# Inheritance in an OSCAL CRM

- ▶ The **leveraged system's** CRM can represent components from the system even if there is no customer responsibility.
- ▶ While individual component references are preferred, if the **leveraged system's** owner or ISSO does not wish to expose individual components, they may still provide a CRM with a "this system" component.
- ▶ Whether individual components or simply a "this system" component, the **leveraged system's** CRM can cite each control satisfied by the component, and provide a customer-appropriate description of the satisfaction.
  - ▶ For example, FedRAMP requires the leveraging system to only describe what is being inherited from a **leveraged system** in satisfaction of a control, but does not require a description of "how" in this case. The CRM can provide a control-statement-specific description of what is being inherited.

